Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

PNA

Probability, Networks and Algorithms

Probability, Networks and Algorithms

An image retrieval system based on adaptive wavelet lifting

P.J. Oonincx, P.M. de Zeeuw

REPORT PNA-R0208 MARCH 31, 2002

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3711

An Image Retrieval System Based on Adaptive Wavelet Lifting

P.J. Oonincx, P.M. de Zeeuw
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
{Patrick.Oonincx,Paul.de.Zeeuw}@cwi.nl

ABSTRACT

We propose an algorithm for content-based image retrieval of grayscale images of objects against a background of texture. It employs feature vectors based on moment invariants of detail coefficients produced by the lifting scheme. The prediction filters in this scheme are chosen adaptively: low order (small stencils) near edges and high order elsewhere. The aim is an algorithm that retrieves similar images of an object irrespective of translation, rotation, reflection or re-sizing of the object, lighting conditions and the background texture. We discuss and propose both linear and nonlinear transforms of the feature vectors in order to develop a suitable metric for measuring the distance between them. We present preliminary results with respect to an artificial database.

2000 Mathematics Subject Classification: 42C40, 62H35, 62P35, 65D05, 94A08
Keywords and Phrases: Image retrieval, lifting scheme, Neville filters, moment invariants, adaptive scheme, feature vectors, affine transformations
Note: This research was supported by the Dutch Technology Foundation (STW), projectnumber CWI.4616

1. INTRODUCTION

Content-based image retrieval (CBIR) is a widely used term to indicate the process of retrieving desired images from a large collection on the basis of features. The extraction process should be automatic (i.e. no human interference) and the features used for retrieval can be either primitive (color, shape, texture) or semantic (involving identity and meaning). In this paper we confine ourselves to grayscale images of objects against a background of texture. This class of images occurs for example in various databases created for the combat of crime: stolen objects [21], tyre tracks and shoe sole impressions [1]. In this report we restrict ourselves to the following problem. Given an image of an object (a so-called query) we want to identify all images in a database which contain the same object irrespective of translation, rotation or re-sizing of the object, lighting conditions and the background texture.

One of the most classical approaches to the problem of recognition of similar images is by the use of moment invariants [11]. This method is based on calculating moments in both the x- and y-direction of the image density function up to a certain order. Hu [11] has shown that certain homogeneous polynomials of these moments can be used as statistical quantities that attain the same values for images that are of the same class, i.e., that can be obtained by transforming one single original image (affine transforms and scaling). However, this method uses the fact that such images consists of a crisp object against a neutral background. If the background contains 'information' (noise, environment in a picture) the background should be the same for all images in one class and should also be obtained from one background using the same transformations. In general this will not be the case. The kind of databases we consider in this paper consists of classes of different objects pasted on different background textures.

To deal with the problem of different backgrounds one may use some filtering process as a preprocessing step. In Do et al. [7] the wavelet transform modulus maxima is used as such preprocessing step. To measure the (dis)similarity between images, moments of the set of maxima points are determined (per scale) and subsequently Hu's invariants are computed. Thus, each image is indexed by a vector in the wavelet maxima moment space. By its construction, this vector predominantly represents shapes.

In this report we propose to bring in adaptivity by using different wavelet filters for smooth and unsmooth parts of the image. The filters are used in the context of the (redundant) lifting scheme [18]. The degree

of ”smoothness” is determined by measuring the *relative local variance* (RLV), which indicates whether locally an image behaves smoothly or not. Near edges low order prediction filters are activated which lead to large lifting detail coefficients along thin curves. At backgrounds of texture high order prediction filters are activated which lead to negligible detail coefficients. Moments and subsequently moment invariants are computed with respect to these wavelet detail coefficients. With the computation of the detail coefficients a certain preprocessing is required to make the method robust for shifts over a non-integer number of gridpoints. Further we introduce *the homogeneity condition* which means that we demand a homogeneous change in the elements of a feature vector if the image seen as a density distribution is multiplied by a scalar.

The report is organized as follows. In Sections 2 and 3 we discuss the lifting scheme and its adaptive version. Section 4 is devoted to the topic of affine invariances of the coefficients obtained from the lifting scheme. In Section 5 the method of moment invariants is recapitulated. The homogeneity condition is introduced which leads to a normalization. Furthermore, the mathematical consequences for the computation of moments of functions represented by fields of wavelet (detail) coefficients are investigated. Section 6 discusses various aspects of the final retrieval algorithm, including possible metrics. Numerical results of the algorithm for a synthetic database are presented in Section 7. Finally, some conclusions are drawn in Section 8.

2. THE LIFTING SCHEME

The lifting scheme as introduced by Sweldens in 1997, see [18], is a method for constructing wavelet transforms that are not necessarily based on dilates and translates of one function. In fact the construction does not rely on the Fourier transform which makes it also suitable for functions on irregular grids. The transform also allows a fully in-place calculation, which means that no auxiliary memory is needed for the computations.

The idea of lifting is based on splitting a given set of data into two subsets. In the one-dimensional case this can mean that starting with a signal $x \in l^2(\mathbb{Z})$ the even and odd samples are collected into two new signals, i.e., $x = x_e + x_o$, where $x_e(n) = x(2n)$ and $x_o(n) = x(2n+1)$, for all $n \in \mathbb{Z}$. The next step of the lifting scheme is to predict the value of $x_e(n)$ given the sequence x_o. This prediction uses a prediction operator $\mathcal{P}$ acting on x_o. The predicted value $(\mathcal{P}x_o)(n)$ is subtracted from $x_e(n)$ yielding a ‘detail’ signal d. An update of the odd samples x_o is needed to avoid aliassing problems. This update is performed by adding $\mathcal{U}d$ to the sequence x_o, with $\mathcal{U}$ the update operator. The lifting procedure can also be seen as a 2-band filter bank. This idea has been depicted in Figure 1. The inverse lifting scheme can

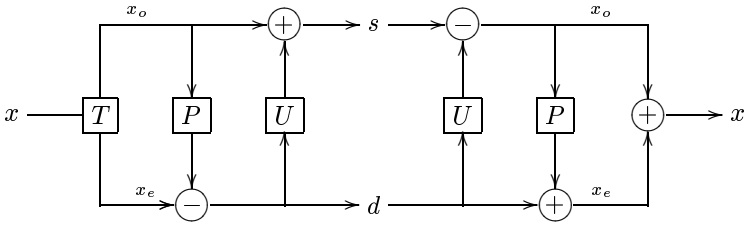


Figure 1: The lifting scheme: splitting, predicting, updating.

immediately be found by undoing the prediction and update operators. In practice, this comes down in Figure 1 to simply changing each $+$ into a $-$ and vice versa.

Compared to the traditional wavelet transform the sequence $d = x_e - \mathcal{P}(x_o)$ can be regarded as detail coefficients of the signal x. The updated sequence $s = x_0 + \mathcal{U}(d)$ can be regarded as the approximation of x at a coarse scale. Using s again as input for the lifting scheme yields detail and approximation signals at lower resolution levels. We observe that every discrete wavelet transform can also be decomposed into a finite sequence of lifting steps [6].

To understand the notion of vanishing moments in terms of the prediction and update operators, we compare the lifting scheme with a two-channel filter bank with analysis filters $\tilde{H}$ (lowpass) and $\tilde{G}$ (highpass) and synthesis filters H and G. Such a filter bank has been depicted in Figure 2. Traditionally we say that a

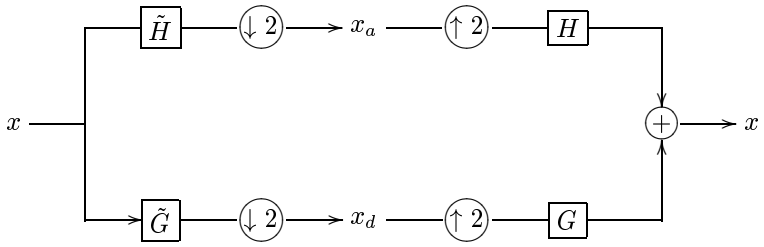


Figure 2: Classical 2-band analysis/synthesis filter bank.

filter bank has N primal and $\tilde{N}$ vanishing moments if

$$Gp = 0,\ p \in \Pi_N \ \text{ and } \ \tilde{G}\tilde{p} = 0,\ \tilde{p} \in \Pi_{\tilde{N}},$$

where Π_N denotes the space of all polynomial sequences of order N. Given the filter operators $\mathcal{P}$ and $\mathcal{U}$, the corresponding filters H and G can be computed by

$$\begin{pmatrix} H(2n) & H(2n+1) \\ G(2n) & G(2n+1) \end{pmatrix} = \begin{pmatrix} 1 & P^*(n) \\ -U^*(n) & 1 - U^*P^*(n) \end{pmatrix}, \tag{2.1}$$

$$\begin{pmatrix} \tilde{H}(2n) & \tilde{H}(2n+1) \\ \tilde{G}(2n) & \tilde{G}(2n+1) \end{pmatrix} = \begin{pmatrix} 1 - UP(n) & U(n) \\ -P(n) & 1 \end{pmatrix}, \tag{2.2}$$

where P and U denote the filter sequences of the operators $\mathcal{P}$ and $\mathcal{U}$ respectively. In [12] Kovacevic and Sweldens showed that we can always use lifting schemes with N primal vanishing moments and $\tilde{N}$ dual vanishing moments ($N \leq \tilde{N}$) by taking for $\mathcal{P}$ a Neville filter of order $\tilde{N}$ with a shift τ and for $\mathcal{U}$ half the adjoint of a Neville filter of order N and shift τ, see [17].

Example 2.1 We take $\mathcal{P} = I$ and $\mathcal{U} = 1/2$. With these operators we get

$$\begin{aligned} s &= x_e/2 + x_o/2, \\ d &= x_e - x_o. \end{aligned}$$

The filter bank has only one vanishing moment ($N = \tilde{N} = 1$). The lifting transform corresponds in this example to the Haar wavelet transform.

Example 2.2 For more vanishing moments, i.e., smoother approximation signals, we take

$$\begin{aligned} (\mathcal{P}x)(n) &= [-x(n+1) + 9x(n) + 9x(n-1) - x(n-2)]/16, \\ (\mathcal{U}x)(n) &= x(n+1)/4 + x(n)/4. \end{aligned}$$

These Neville filters give rise to a 2-channel filter bank with 2 primal and 4 dual vanishing moments.

The lifting scheme can also be used for higher dimensional signals $x \in l^2(\mathbb{Z}^n)$. For these signals the lifting scheme consists of M channels, where M denotes the absolute value of the determinant of the dilation matrix, that is used in the corresponding discrete wavelet transform. In each channel the signal is translated along one of the M coset representatives from the unit cell of the corresponding lattice, see [12]. The signal in the first channel is then used for predicting the data in all other channels by using $M - 1$ possible different prediction operators $\mathcal{P}_1, \ldots, \mathcal{P}_{M-1}$. Thereafter the first channel is updated using update operators $\mathcal{U}_1, \ldots, \mathcal{U}_{M-1}$ on the $M - 1$ other channels.

Let us consider an image as a two-dimensional signal. An important example of the lifting scheme applied to such a signal is one that involves 2 channels ($M = 2$). We subdivide the lattice on which the signal has been defined into two sets on quincunx grids, see Figure 3. This division is also called "checkerboard" or "red-black" division. The pixels on the red spots (○) are used to predict the samples on the black spots (●), while updating of the red spots is performed by using the detailed data on the black spots. An example of a lifting transform with second order prediction and update filters is given by

$$\begin{aligned} (\mathcal{P}x)(i,j) &= [x(i-1,j) + x(i,j-1) + x(i+1,j) + x(i,j+1)]/4,\ i \bmod 2 \neq j \bmod 2, \\ (\mathcal{U}x)(i,j) &= [x(i-1,j) + x(i,j-1) + x(i+1,j) + x(i,j+1)]/8,\ i \bmod 2 = j \bmod 2. \end{aligned}$$

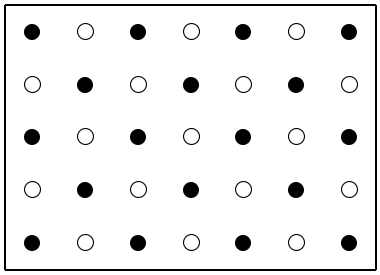


Figure 3: A rectangular grid composed of two quincunx grids.

order N	V_1	V_2	V_3	V_4	V_5	V_6	V_7
2	$1/4$	0	0	0	0	0	0
4	$10/32$	$-1/32$	0	0	0	0	0
6	$87/2^8$	$-27/2^9$	2^{-8}	$3/2^9$	0	0	0
8	$5825/2^{14}$	$-2235/2^{15}$	$625/2^{16}$	$425/2^{15}$	$-75/2^{16}$	$9/2^{16}$	$-5/2^{12}$

Table 1: Quincunx Neville filter coefficients

The algorithm using the quincunx lattice is also known as the red-black wavelet transform by Uytterhoeven and Bultheel, see [20]. In general $\mathcal{P}$ can be written as

$$(\mathcal{P}x)(i,j) = \sum_{(n,m)\in S_{\bar{N}}} a_{\bar{N}}(n,m)\, x(i+n, j+m),\ i\, mod\, 2 \neq j\, mod\, 2, \tag{2.3}$$

with $S_{\bar{N}}$ a subset of $\{(n,m) \in \mathbb{Z}^2 \mid (n+m) \bmod 2 = 1\}$ and $a_{\bar{N}}(s)$, $s \in S_{\bar{N}}$, a set of coefficients in $I\!R$. In this case a general formula for $\mathcal{U}$ reads

$$(\mathcal{U}x)(i,j) = \sum_{(n,m)\in S_N} a_N(n,m)\, x(i+n, j+m)/2,\ i\, mod\, 2 = j\, mod\, 2, \tag{2.4}$$

with S_N depending on the number of required primal vanishing moments N. For several elements in S_N the coefficients $a_N(s)$ attain the same values. Therefore we take these elements together in subsets of S_N, i.e.,

$$\begin{aligned} V_1 &= \{(\pm 1, 0), (0, \pm 1)\},\ V_2 = \{(\pm 1, \pm 2), (\pm 2, \pm 1)\},\ V_3 = \{(\pm 3, 0), (0, \pm 3)\}, \\ V_4 &= \{(\pm 2, \pm 3), (\pm 3, \pm 2)\},\ V_5 = \{(\pm 1, \pm 4), (\pm 4, \pm 1)\},\ V_6 = \{(\pm 5, 0), (0, \pm 5)\}, \\ V_7 &= \{(\pm 3, \pm 4), (\pm 4, \pm 3)\} \end{aligned} \tag{2.5}$$

Table 1 indicates the values of all $a_N(s)$, $s \in V_k$, for different values of N (2 through 8) when using quincunx Neville filters, see [12], which are the filters we use in our approach. We observe that $S_8 = V_1 + \cdots + V_7$ and so a 44 taps filter is used as prediction/update if the required filter order is 8. For an illustration of the Neville filter of order 4 see Figure 4. Here the numbers $1, 2$ correspond to the values of the filter coefficients as given in V_1 and V_2 respectively at that position. The left-hand filter can be used to transform a signal defined on a quincunx grid into a signal defined on a rectangular grid, the right-hand filter is the 45 degrees rotated version of the left-hand filter and can be used to transform a signal from a rectangular grid towards a quincunx grid.

We observe that the quincunx lattice yields a non separable 2D-wavelet transform, which is also symmetric in both horizontal and vertical direction. Furthermore, we only need one prediction and one update operator for this 2D-lifting scheme, which reduces the number of computations.

The prediction and update operators for the quincunx lattice do also appear in schemes for other lattices, like the standard 2D-separable lattice and the hexagonal lattice [12]. The algorithm for the quincunx lattice can be extended in a rather straightforward way for these two other well-known lattices.

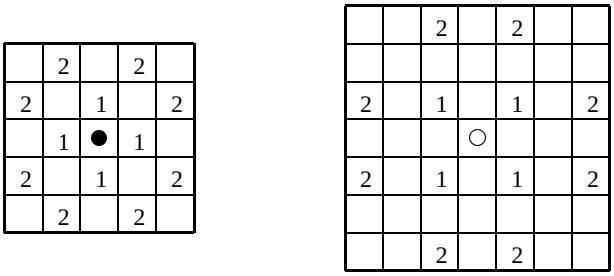


Figure 4: Neville filter of order 4: rectangular (left) and quincunx (right)

Figure 5 illustrates the possibility of the use of more than 2 channels in the lifting scheme. Here $M = 4$ channels are employed, using a four-colour division of the 2D-lattice. It involves $M - 1 = 3$ (interchangeable) prediction steps. Each of the subsets with colours $\circ$, $\triangleright$ and $\square$ respectively, is predicted by application of a prediction filter on the subset with colour $\bullet$.

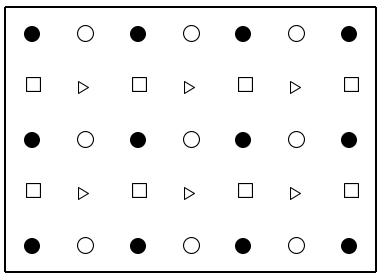


Figure 5: Separable grid (four-colour division).

3. Adaptive Lifting

When using the lifting scheme or a classical wavelet approach, the prediction/update filters or wavelet/scaling functions are chosen in a fixed fashion. Generally they can be chosen in such way that a signal is approximated with very high accuracy using only a limited number of coefficients. Discontinuities mostly give rise to large detail coefficients which is undesirable for applications like compression. For our purpose large detail coefficients near edges in an images are desirable, since they can be identified with the shape of objects we want to detect. However, they are undesirable if such large coefficients are related to the background of the image. This situation occurs if a small filter is used on a background of texture that contains irregularities locally. In this case a large smoothing filter gives rise to small coefficients for the background.

These considerations lead to the idea of using different prediction filters for different parts of the signal. The signal itself should indicate (for example by means of local behavior information) whether a high or low order prediction filter should be used. Such an approach is commonly referred to as an adaptive approach. Many of these adaptive approaches have been described already thoroughly in the literature, e.g. [3, 4, 8, 13, 19]. In this paper we follow the approach proposed by Baraniuk et al. in [2], called the space-adaptive approach. This approach follows the scheme as shown in Figure 6.

After splitting all pixels of a given image I into two complementary groups I_r and I_b (red/black), the pixels in I_r are used to predict the values in I_b. This is done by means of a prediction filter acting on I_r, i.e., $\mathcal{P}(I_r)$. In the adaptive lifting case this prediction filter depends on local information of the image pixels I_r. Choices for $\mathcal{P}$ may vary from high to low order filters, depending on the regularity of the image locally. For the update operator, we choose the update filter that corresponds to the prediction filter with lowest order from all possible to be chosen $\mathcal{P}$. The order of the update filter should be lower or equal to the order of the prediction filter as a condition to provide a perfect reconstruction filter bank. As with the classical wavelet filter bank approach, the order of the prediction filter equals the number of dual vanishing

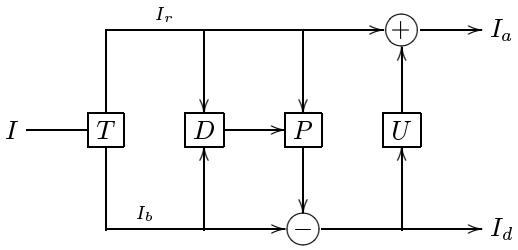


Figure 6: Generating coefficients via adaptive lifting

moments while the order of the update filter equals the number of primal vanishing moments, see [12]. The above leads us to use a second order Neville filter for the update step and an Nth order Neville filter for the prediction step, where $N \in \{2, 4, 6, 8\}$. In our application the reconstruction part of the lifting scheme is not needed.

In [2], Baraniuk et al. choose to start the lifting scheme with an update operator $\mathcal{U}$ followed by an adaptively chosen prediction operator. The reason for interchanging the prediction and update operator is that this solves stability and synchronization problems in lossy coding applications. We will not discuss this topic in further detail, but only mention that they took for the filters of $\mathcal{U}$ and $\mathcal{P}$ the $(1, N)$ branch of the Cohen-Daubechies-Feauveau (CDF) filter family [5]. The order of the prediction filter N was chosen to be 1, 3, 5 or 7, depending on the local behavior of the signal. The filter orders of the CDF filters in their paper correspond to the filter orders of the Neville filters we are using in our approach.

Relative local variance We propose a measure on which the decision operator in the 2D adaptive lifting scheme can be based on, namely the *relative local variance* of an image. This relative local variance (RLV) of an image I is given by

$$\text{rlv}[I](i,j) \quad = \quad \sum_{k=i-T}^{i+T} \sum_{l=j-T}^{j+T} (I(k,l) - \overline{\mu_{i,j}})^2 / \text{var}\,(I), \tag{3.1}$$

with

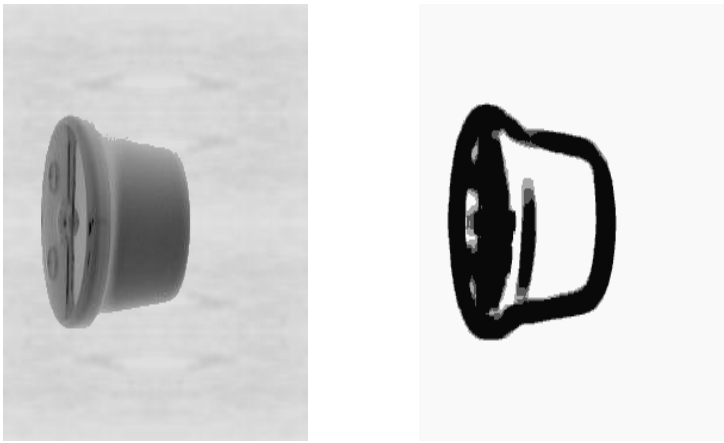
$$\overline{\mu_{i,j}} \quad = \quad \sum_{k=i-T}^{i+T} \sum_{l=j-T}^{j+T} I(k,l)/(2T+1)^2. \tag{3.2}$$

For the window size we take $T = 5$, since with this choice all $I(k, l)$ that are used for the prediction of $I(i, j)$ contribute to the RLV for (i, j), even for the 8th order Neville filter. When the RLV is used at higher resolution levels we first have to down sample the image I appropriately.

The first time the prediction filter is applied (to the upper left pixel) we use the 8th order Neville filter on the quincunx lattice as given in Table 1. For all other subsequent pixels (i, j) to be predicted, we first compute $\text{rlv}[I](i, j)$. Then quantizing the values of the RLV yields a decisionmap indicating which prediction filter should be used at which positions. Values above the highest quantizing level induce a 2nd order Neville filter, while values below the lowest quantizing levels induce an 8th order Neville filter. For the quantizing levels we take multiples of the mean of the RLV. Test results have shown that $[\mu(\text{rlv}) \quad 1.5\,\mu(\text{rlv}) \quad 2\,\mu(\text{rlv})]$ are quantizing levels that yield a good performance in our application. In Figure 7 we have depicted an image (left) and its decision map based on the RLV (right).

4. Affine Invariant Lifting

Although both traditional wavelet analysis and the lifting scheme yield detail and approximation coefficients that are localised in scale and space, they are both not translation invariant. This means that if a signal or image is translated along the grid, its lifting coefficients may not be just be given by a translation of the original coefficients. Moreover, in general the coefficients will attain values in the same range of the original values (after translation), but they will be totally different.



a) original image b) decision map (RLV)

Figure 7: An object on a wooden background and its rel. local variance (decision map): white=8th order, black=2nd order.

For studying lifting coefficients of images a desirable property would also be invariance under reflections and rotations. However, for these two transformations we have to assume first that the values of the image on the grid points is not affected a rotation or reflection. In practice, this means that we only consider reflections in the horizontal, the vertical and the diagonal axis and rotations over multiples of $\pi/2$.

4.1 Redundant Lifting

For the classical wavelet transform a solution for translation invariance is given by the redundant wavelet transform [15], which is a non-decimated wavelet (at all scales) transform. This means that one gets rid of the decimation step. As a consequence the data in all subbands have the same size as the size as the input data of the transform. Furthermore, at each scaling level, we have to use zero padding to the filters in order to keep the multiresolution analysis consistent. Not only more memory is used by the redundant transform, also the computing complexity of the fast transform increases. For the non-decimated transform computing complexity is $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N)$ for the fast wavelet transform.

Whether the described redundant transform is also invariant under reflections and rotations depends strongly on the filters (wavelets) themselves. Symmetry of the filters is necessary to guarantee certain rotation and reflection invariances. This is a condition that is not satisfied by many well-known wavelet filters.

The redundant wavelet transform can also be translated into a redundant lifting scheme. In one dimension this works out as follows. Instead of partitioning a signal $x \in l^2(\mathbb{Z})$ into x_e and x_o we copy x to both x_e and x_o. The next step of the lifting scheme is to predict x_e by

$$x_e = x_e - \mathcal{P}^{(j)} x_o. \tag{4.1}$$

The prediction filter $\mathcal{P}^{(j)}$ is the same filter as used for the non-redundant case, however now it depends on the resolution level, since at each level zero padding is applied to $\mathcal{P}$. This holds also for the update filters $\mathcal{U}^{(j)}$. So, the update step reads

$$x_o = x_o + \mathcal{U}^{(j)} x_e. \tag{4.2}$$

For higher dimensional signals $x \in l^2(\mathbb{Z}^n)$ we copy the data in all M channels of the used filter bank. Next the M-channel lifting scheme is applied on the data, using zero padding for the filters at each resolu-

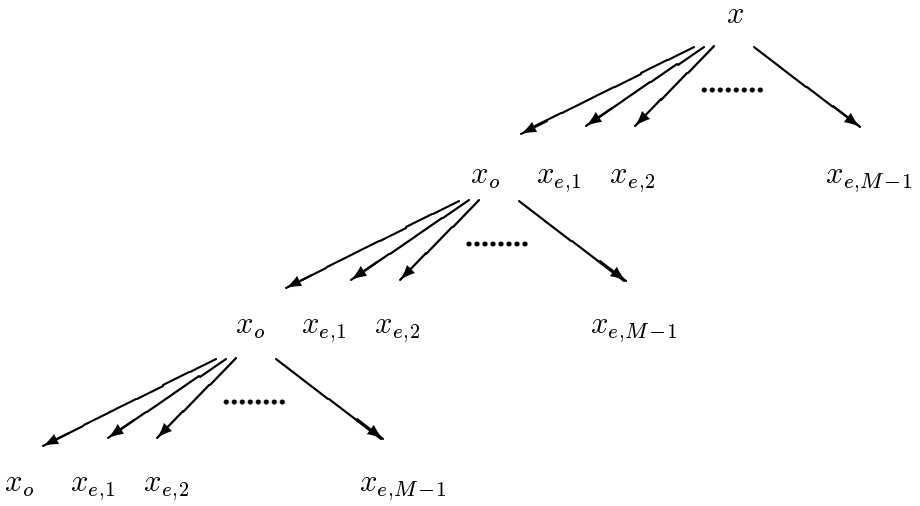


Figure 8: Tree structure of the M-channel lifting scheme.

tion level. Remark, that for each lifting step in the redundant M-channel lifting scheme we have to store at each scaling level M times as much data as in the non-redundant scheme, see Figure 8.

We observe that in our approach Neville filters on a quincunx lattice are used. Due to their symmetry properties, see Table 1, the redundant scheme does not only guarantee translation invariance, but also invariance under rotations over multiples of $\pi/2$ and reflections in the horizontal, vertical and diagonal axis is assured. Invariance under other rotations and reflections can not be guaranteed by any prediction and update filter pair, since the quincunx lattice is not invariant under these transformations.

4.2 An Attempt to Avoid Redundancy: Fixed Point Lifting

As we have seen the redundant scheme provides a way of finding detail and approximation coefficients that are invariant under translations, reflections and rotations, under which the lattice is also invariant. Due to its redundancy this scheme is stable in the sense that it treats all samples of a given signal in the same way. However redundancy also means additional computational costs and perhaps even worse additional memory to store the coefficients. Therefore we started searching for alternative schemes that are also invariant under the described class of affine transformation. Although we did not yet manage to come up with an efficient stable scheme, we would like to stretch the principal idea behind the building blocks of such approach. In the sequel we will only use the redundant lifting scheme as described in the preceding section.

Before we start looking for possible alternative schemes we examine why the lifting scheme is not translation invariant. Assume we have a signal $x \in l^2(\mathbb{Z}^n)$ that is analysed with an M-band lifting scheme. Then after one lifting step we have approximation data x_e and detail data $x_{o,1}, \ldots, x_{o,M-1}$. Whether one sample $x(i)$, $i \in \mathbb{Z}^n$, is determined to become either a sample of x_e or a sample of $x_{o,1}, \ldots, x_{o,M-1}$ depends only on its position on the lattice and the way we partition the lattice into M groups. Of course, this partitioning is rather arbitrary. The more channels we use the higher the probability is that for a fixed partitioning one sample $x(i)$ that was determined to be used for predicting other samples, will become a sample of $x_{o,1}, \ldots, x_{o,M-1}$ after translating x.

Following Figure 8 it is clear that any sample $x(i)$, $i \in \mathbb{Z}^n$, can end up after J lifting steps in $J(M-1)+1$ ways, either in approximation data at level J or in detail data at some level $1, 2, \ldots, J$.

The idea of the alternative scheme we propose here is to partition a signal not upon its position on the lattice but upon its structure. This means that for each individual signal we indicate a fixed point for which we demand that it will end up in the approximation data after J lifting steps. If this point can be chosen independent of its coordinates on the lattice, the lifting scheme based on this partitioning will then translation invariant. For higher dimensional signals we can also achieve invariance under the other discussed affine transformations, however then we have to fix more points, depending on the number of

channels. In our approach the quincunx lattice is used and therefore fixing one approximation sample on J scales will immediately fix the partitioning of all other samples on the quincunx lattice at scale $1, \ldots, J$. As a result the fixed point lifting scheme is invariant under translations, rotations and reflections that leave the quincunx lattice invariant. In the sequel of this chapter we will only discuss the lifting scheme for $x \in l^2(\mathbb{Z}^2)$ for the quincunx lattice.

Although the proposed fixed point lifting scheme may seem to be a powerful tool for affine invariant lifting, it will be hard to deal with in practice. The problem we will have to face is how to choose a fixed point in every image $x \in l^2(\mathbb{Z}^2)$. In other words we have to find a suitable decision operator $\mathcal{S}$ that adds to every $x \in l^2(\mathbb{Z}^2)$ a unique $i_p \in \mathbb{Z}^2$, its fixed point, i.e.,

$$\mathcal{S}(x) = i_p.$$

If we demand $\mathcal{S}$ to depend only on x and not on the lattice (coordinate free) it will be hard to find such $\mathcal{S}$ that is well defined. This independence of the coordinates is necessary for rotation invariances. However, this is not the only difficulty we have to face.

Stability of the scheme is an other problem. If for some reason a fixed point i'_p has been wrongly indicated, for example due to truncation errors, the whole scheme might collapse down. Although we cannot easily solve the problem of determining incorrect fixed points i'_p we can increase the stability of the scheme by not imposing that at each scale i_p should be an index number of the coarse scale data after zero padding. Instead of this procedure we rather determine a fixed point for both the original signal x (i_p) and for the coarse scale data $x_o^{(j)}$ ($i_p^{(j)}$) at each scale $j = 1, \ldots, J$. Then we impose that $i_p^{(j)}$ should be used for prediction in the $j+1$th lifting step, for $j = 0, \ldots, J-1$ and with $i_p^{(0)} = i_p$. Furthermore, stability may be increased by using decision operators that generate a set of fixed points. However, since no stable method (uniform decision operator) is available yet, we will use the redundant lifting scheme in our approach and do not work out the idea of fixed point lifting here at this moment.

5. Moment Invariants

At the outset of this section we give a brief introduction into the theory of statistical invariants for imaging purposes, based on centralized moments. Traditionally, these features have been widely used in pattern recognition applications to recognize the geometrical shapes of different objects [11]. Here, we will compute invariants with respect to the detail coefficients as produced by the wavelet lifting schemes of Sections 2– 4. We use invariants based on moments of the coefficients up to third order. We show how to construct a feature vector from the obtained wavelet coefficients at several scales It is followed by proposals for normalization of the moments to keep them in comparable range.

5.1 Introduction and recapitulation

We regard an image as a density distribution function $f \in S(I\!R^2)$, the Schwartz class. In order to obtain translation invariant statistics of such f we use central moments of f for our features. The $(p+q)th$ order central moment $\mu_{pq}(f)$ of f is given by

$$\mu_{pq}(f) = \int_{I\!R} \int_{I\!R} (x - x_c)^p (y - y_c)^q f(x, y)\, d(x - x_c)\, d(y - y_c), \tag{5.1}$$

with the center of mass

$$x_c = \frac{\int_{I\!R} \int_{I\!R} x f(x, y)\, dx\, dy}{\int_{I\!R} \int_{I\!R} f(x, y)\, dx\, dy} \quad \text{and} \quad y_c = \frac{\int_{I\!R} \int_{I\!R} y f(x, y)\, dx\, dy}{\int_{I\!R} \int_{I\!R} f(x, y)\, dx\, dy}. \tag{5.2}$$

Computing the centers of mass x'_c and y'_c of $g(x,y) = f(x-a, y-b)$ yields

$$\begin{aligned} x'_c &= \frac{\int_{I\!R}\int_{I\!R} x f(x-a,y-b)\,dx\,dy}{\int_{I\!R}\int_{I\!R} f(x-a,y-b)\,dx\,dy} \\ &= \frac{\int_{I\!R}\int_{I\!R} (x+a) f(x,y)\,dx\,dy}{\int_{I\!R}\int_{I\!R} f(x,y)\,dx\,dy} = x_c + a, \end{aligned}$$

and similarly $y'_c = y_c + b$. Combining this with (5.1) shows

$$\begin{aligned} \mu_{pq}(g) &= \int_{I\!R}\int_{I\!R} (x-x'_c)^p (y-y'_c)^q f(x-a,y-b)\,d(x-x'_c)\,d(y-y'_c) \\ &= \int_{I\!R}\int_{I\!R} (x+a-x_c-a)^p (y+b-y_c-b)^q f(x,y)\,d(x+a-x_c-a)\,d(y+b-y_c-b) \\ &= \int_{I\!R}\int_{I\!R} (x-x_c)^p (y-y_c)^q f(x,y)\,d(x-x_c)\,d(y-y_c) = \mu_{pq}(f), \end{aligned}$$

i.e., the central moments are translation invariant.

We also require that the features should be invariant under orthogonal transformations (rotations). For deriving these features we follow [11] using a method with homogeneous polynomials h of order r. These are given by

$$h[c_{r,0}, c_{r-1,1}, \ldots, c_{1,r-1}, c_{0,r}](u,v) = \sum_{k=0}^{r} c_{r-k,k} \binom{r}{k} u^{r-k} v^k. \tag{5.3}$$

Now assume that the variables $u, v \in I\!R$ are obtained from other variables $u', v' \in I\!R$ under some linear transformation $A : I\!R^2 \to I\!R^2$, i.e.,

$$(u,v)^T = A\,(u',v')^T,$$

then h is an algebraic invariant of weight w if

$$h[c'_{r,0}, c'_{r-1,1}, \ldots, c'_{1,r-1}, c'_{0,r}] = \det(A)^w\, h[c_{r,0}, c_{r-1,1}, \ldots, c_{1,r-1}, c_{0,r}]. \tag{5.4}$$

with $c'_{n,m}$ the new coefficients obtained after transforming h by A.

For orthogonal transformations A we have $\det(A) = 1$ and therefore h is invariant under rotations if

$$h[c'_{r,0}, c'_{r-1,1}, \ldots, c'_{1,r-1}, c'_{0,r}] = h[c_{r,0}, c_{r-1,1}, \ldots, c_{1,r-1}, c_{0,r}].$$

Particularly we have from [11], that if $h[c_{r,0}, c_{r-1,1}, \ldots, c_{1,r-1}, c_{0,r}]$ is an algebraic invariant, then also the moments of order r have the same invariant, i.e.,

$$h[\mu'_{r,0}, \mu'_{r-1,1}, \ldots, \mu'_{1,r-1}, \mu'_{0,r}] = h[\mu_{r,0}, \mu_{r-1,1}, \ldots, \mu_{1,r-1}, \mu_{0,r}]. \tag{5.5}$$

From this equation 2 functions of second order can be derived that are invariant under rotations, see [11]. For $r = 2$ we have the invariants

$$I_1 = \mu_{20} + \mu_{02} \;\text{ and }\; I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2.$$

It was also shown that these two functions are also invariant under reflections, which can be a useful property for identifying reflected images.

Since the way of deriving these invariants may seem a bit technical and artificial, we illustrate with straightforward calculus that I_1 and I_2 are indeed invariant under rotations. The invariance under reflections is left to the reader, since showing this follows the same calculations.

We consider the rotated distribution function

$$f_\theta(x,y) = f(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta).$$

and the corresponding invariants $I_1(\theta)$ and $I_2(\theta)$, which are I_1 and I_2 but now based on moments calculated from f_θ. So what we have to show is that $I_1(\theta) = I_1$ and $I_2(\theta) = I_2$. It follows from (5.1) and (5.2) that $I_1(\theta) = I_1$ if and only if

$$(u\cos\theta + v\sin\theta)^2 + (v\cos\theta - u\sin\theta)^2 = u^2 + v^2,$$

with $u := x - x_c$ and $v = y - y_c$. Obviously this holds true, considering the trigonometric rule $\cos^2\theta + \sin^2\theta = 1$.

To do the same for I_2 we also have to introduce $w := x - x_c$ and $z := y - y_c$. Because we have to take products of integrals that define μ_{pq}, we cannot use u and v in both integrals. As for I_1 we can now derive from (5.1) and (5.2) that $I_2(\theta) = I_2$ if and only if

$$\begin{aligned} & (u^2 - v^2)(w^2 - z^2) + 4uvwz \\ = \; & ((u\cos\theta + v\sin\theta)^2 - (v\cos\theta - u\sin\theta)^2)((w\cos\theta + z\sin\theta)^2 - (z\cos\theta - w\sin\theta)^2) + \\ & 4(u\cos\theta + v\sin\theta)(v\cos\theta - u\sin\theta)(w\cos\theta + z\sin\theta)(z\cos\theta - w\sin\theta). \end{aligned}$$

We simplify the right-hand side term by term. The first term, that is related to $(\mu_{20} - \mu_{02})^2$ becomes

$$((u^2 - v^2)\cos(2\theta) + 2uv\sin(2\theta))((w^2 - z^2)\cos(2\theta) + 2wz\sin(2\theta)).$$

The second term (related to $4\mu_{11}^2$) becomes

$$((u^2 - v^2)\sin(2\theta) - 2uv\cos(2\theta))((w^2 - z^2)\sin(2\theta) - 2wz\cos(2\theta)).$$

Adding these two terms gives us

$$\begin{aligned} & (u^2 - v^2)(w^2 - z^2)(\cos^2(2\theta) + \sin^2(2\theta)) + 4uvwz(\cos^2(2\theta) + \sin^2(2\theta) \\ = \; & (u^2 - v^2)(w^2 - z^2) + 4uvwz, \end{aligned}$$

which demonstrates that indeed also I_2 is invariant under rotations. Similar calculus shows that invariance under reflections also holds.

From Equation (5.5) four functions of third order and one function of both second and third order can be derived that are invariant under both rotations and reflecting, namely

$$I_3 = p_1^2 + p_2^2,\ I_4 = p_3^2 + p_4^2 \ \text{ and } \ I_5 = 2\left(p_1p_3^3 - 3p_1p_3p_4^2 + 3p_1p_3^2p_4 - p_1p_4^3\right),$$

with

$$p_1 = \mu_{30} - 3\mu_{12},\ p_2 = \mu_{03} - 3\mu_{21},\ p_3 = \mu_{30} + \mu_{12},\ p_4 = \mu_{03} + \mu_{21}.$$

The last polynomial that is invariant under both rotations and reflections consists of both second and third order moments and is given by

$$I_6 = 2p_5p_3^2 - 2p_5p_4^2 + 8\mu_{11}p_3p_4,$$

with $p_5 = \mu_{20} - \mu_{02}$ and p_3, p_4 as above. To these six invariants we can add a seventh one, which is only invariant under rotations and changes sign under reflections. It is given by

$$I_7 = 2\left(p_1p_4^3 + 3p_2p_3p_4^2 - 3p_1p_3^2p_4 - p_2p_3^3\right).$$

Since we want to include reflections as well in our set of invariant transformations we will use $|I_7|$ instead of I_7 in our approach. From now on, we will identify $|I_7|$ with I_7.

We observe that all possible linear combinations of these invariants are invariant under proper orthogonal transformations and translations. Therefore we can call these seven invariants also invariant generators.

5.2 Normalization

Here we increase the number of circumstances under which images should be considered invariant. Firstly, if an image consists of a crisp object against a neutral background, a change in the dimensions of the object should still lead to the conclusion that the object remains the same. Invariance under this type of scaling, the similitude invariance, can be obtained by normalizing the moments μ_{pq}. Secondly, differences in luminosity also affect images of the same object. We will allow for a moderate variance with varying luminosity (the so-called homogeneity condition).

Similitude invariance Uniform dilations (by a scalar $\alpha > 0$) of the whole image or objects in an image against a neutral background will result in new central moments given by[11]

$$\mu'_{pq} = \alpha^{p+q+2}\mu_{pq}. \tag{5.6}$$

It follows in particular that $\mu'_{00} = \alpha^2\mu_{00}$, and also $\mu'_{20} + \mu'_{02} = \alpha^4(\mu_{20} + \mu_{02})$. Combining this result with (5.6) yields

$$\frac{\mu'_{pq}}{(\mu'_{00})^{\frac{p+q+2}{2}}} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q+2}{2}}} \quad \text{and} \quad \frac{\mu'_{pq}}{(\mu'_{20}+\mu'_{02})^{\frac{p+q+2}{4}}} = \frac{\mu_{pq}}{(\mu_{20}+\mu_{02})^{\frac{p+q+2}{4}}}$$

respectively. As we recall that both μ_{00} and $\mu_{20} + \mu_{02}$ are invariants w.r.t. rotation and reflection this demonstrates how to normalize the moments to achieve invariance under dilation. The first choice leads to the following new set of invariant generators

$$I'_1 = I_1/\mu_{00}^2,\ I'_2 = I_2/\mu_{00}^4,\ I'_3 = I_3/\mu_{00}^5,\ I'_4 = I_4/\mu_{00}^5,\ I'_5 = I_5/\mu_{00}^{10},\ I'_6 = I_6/\mu_{00}^7,\ I'_7 = I_7/\mu_{00}^{10}. \tag{5.7}$$

The second choice leads to a different but similar result. It may be more suitable (as a starting point) in case the density distribution is given by wavelet detail coefficients (Section 6.3).

The homogeneity condition We introduce a feature vector, derived from the invariants discussed in the previous section. Distances between feature vectors, measured by e.g. the Euclidean norm, are supposed to indicate the difference between, or resemblance of, images. Obviously, the specific choice of the separate elements, e.g. the weights assigned to these (or alternatively employed within a norm), may lead to highly different results. If, in a first and naive approach, we compose the following feature vector $I \in I\!R^7$:

$$I \equiv (I_1 \quad I_2 \quad I_3 \quad I_4 \quad I_5 \quad I_6 \quad I_7)^T \tag{5.8}$$

with the I_k as defined in Section 5.1, then, indeed, hereby obtained results turn out useless as the various elements appear to operate in different orders of magnitude. We remedy the arbitrariness of the definition of the vector by introducing an additional condition: *the homogeneity condition*, to be described and explained below.

Consider the simple model that for a change in luminosity of an object the distribution function f is mapped onto a different f' by means of an affine transformation. For images this means that the grayvalue of all pixels is multiplied by a scalar $\lambda > 0$ and shifted along a distance b (the offset). A scalar multiplication of the distribution function f does not affect the center of mass (5.2). From (5.1) it follows directly that

$$\mu_{pq}(\lambda f) = \lambda\mu_{pq}(f), \text{ for all } \lambda \neq 0. \tag{5.9}$$

The homogeneity condition means that we demand a homogeneous change in the elements of a feature vector if the density distribution f is multiplied by the said scalar. We observe that neither the vector I (5.8) nor the vector I' derived from (5.7), satisfies the homogeneity condition, as multiplication of f by $\lambda \neq 0$ leads to the transforms:

$$I \longrightarrow (\lambda I_1 \quad \lambda^2 I_2 \quad \lambda^2 I_3 \quad \lambda^2 I_4 \quad \lambda^4 I_5 \quad \lambda^3 I_6 \quad \lambda^4 I_7 \quad)^T$$

and

$$I' \longrightarrow (\lambda^{-1}I'_1 \quad \lambda^{-2}I'_2 \quad \lambda^{-3}I'_3 \quad \lambda^{-3}I'_4 \quad \lambda^{-6}I'_5 \quad \lambda^{-4}I'_6 \quad \lambda^{-6}I'_7 \quad)^T.$$

The following operator

$$\mathcal{R}_p(u) = \text{ sign}(u)|u|^{1/p}, \text{ for } p \in I\!N \text{ and } u \in I\!R \tag{5.10}$$

when applied to an invariant I_k produces again an invariant. It is a "legal" operation that invariants can be subjected to, i.e., neither their invariance properties nor their discriminative power are lost. The feature vectors

$$\tilde{I} \quad = \quad (I_1 \quad \mathcal{R}_2(I_2) \quad \mathcal{R}_2(I_3) \quad \mathcal{R}_2(I_4) \quad \mathcal{R}_4(I_5) \quad \mathcal{R}_3(I_6) \quad \mathcal{R}_4(I_7))^T, \tag{5.11}$$

and

$$\tilde{I}' \quad = \quad (I'_1 \quad \mathcal{R}_2(I'_2) \quad \mathcal{R}_3(I'_3) \quad \mathcal{R}_3(I'_4) \quad \mathcal{R}_6(I'_5) \quad \mathcal{R}_4(I'_6) \quad \mathcal{R}_6(I'_7))^T \tag{5.12}$$

now both satisfy the homogeneity condition as can be easily verified. By numerical experiments it is shown that hereby the vector elements remain in comparable range. (At this point we might consider to introduce a distance measure between feature vectors F and F' which vanishes altogether if $F = \lambda F'$, $\lambda \in I\!R$. This is a matter for future investigation.) Next, we discuss the offset.

Offset in the affine transform The detail coefficients produced by the lifting scheme are invariant to the offset b in the affine transform of our luminosity model. To show this we construct a new image $y \in l^2(\mathbb{Z}^2)$, given an image $x \in l^2(\mathbb{Z}^2)$, by

$$y(i,j) = x(i,j) + b,$$

for all $i, j \in \mathbb{Z}$ and for a given $b \in I\!R$. According to (2.3) the detail coefficients of y are given by

$$\begin{aligned} d_y(i,j) &= y(i,j) - (\mathcal{P}y)(i,j) \\ &= x(i,j) + b - \sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m)\,(x(i+n,j+m)+b) \\ &= x(i,j) + b - b - \sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m)\,x(i+n,j+m) \\ &= d_x(i,j), \end{aligned}$$

if and only if

$$\sum_{(n,m)\in S_{\tilde{N}}} a_{\tilde{N}}(n,m) = 1,$$

for the given prediction filter. This necessary and sufficient condition is satisfied for the Neville filters we use in our approach, independent of the filter order. Generally, the detail coefficients are invariant to offsets in the range of an image if and only if the sum of the prediction filter coefficient is equal to 1. Also in classical wavelet analysis this is a well-known and commonly used condition.

The number of coefficients To create feature vectors from a given image, using a wavelet lifting approach, we first compute the detail coefficients d_1 that appear in the lifting scheme for this image. The coarse scale approximation data s_1 is used again in the lifting scheme to obtain d_2 and s_2. After K recursive lifting steps we can use the detail coefficients $d_1, \ldots, d_K$ for computing feature vectors, as follows. After each step we select a relatively small set of coefficients. The set of these coefficients is given by W_j. All other detail coefficients $d_j(n,m)$, $(n,m) \notin W_j$ are put to zero. Of this newly constructed 'image' d_j the moments μ^j_{pq} are computed. However, before calculating the invariants at this scale j we have to take into account that the number of coefficients L_j may not be a constant as a function of j. Therefore we choose to use moments ν_{pq} that are normalized to the number of coefficients as well, i.e., $\nu^j_{pq} = \mu^j_{pq}/L_j$. The criterions to select the coefficients may be based on thresholding, or, which is more sophisticated and reliable, on edge detection, see Section 6.1.

After K lifting steps we have at our disposal $\tilde{I}^{(j)}$, $j = 1, \ldots, K$. Combining them into one $7K$-dimensional vectors yields the feature vector. We take $K = 4$, a common choice with other wavelet approaches [7, 14]. We compute the distance between images as the variance weighted Euclidean distance between two feature vectors. The weight factors are given by the inverse variances for each vector entry, computed over all corresponding entries in the feature vectors resulting from the database we use. For an elaborate description of this metric (and an alternative) see Sections 6.4–6.5.

5.3 Numerical computation of moments

We elaborate briefly on the numerical computation of moments. Figure 9 elucidates the computation of moments on a rectangular grid. Each pixel corresponds to a point in the domain. To each point a value

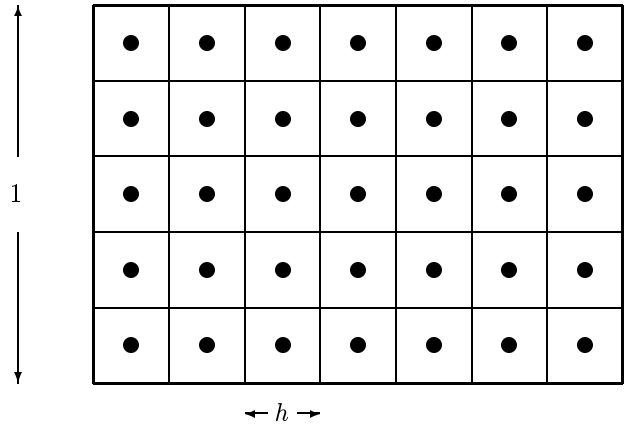


Figure 9: Supports at rectangular grid.

has been associated: this can be grayvalue at a pixel, a wavelet coefficient, etc. Using the values of the image pixels (or lifting coefficients) we construct an interpolating function based on piecewise constant approximation. The piecewise constant basisfunctions have their support on the squares, as indicated in Figure 9. Though the above interpolating function is not in the Schwartz class, it has compact support, is measurable and can be integrated.

A practical assumption is that the shortest side of the rectangular domain has size 1. The actual size (h) of the supports then follows from the number of gridpoints. As the supports are square, the dimension of the longest side of the domain also follows at once. We can now perform the integration in (5.1)–(5.2) numerically and thereby compute the moments.

We note the following. When a function $f \in S(\mathbb{R}^2)$ is approximated piecewise constantly with an increasing number of gridpoints (pixels) then, roughly speaking, the moments remain invariant.

6. RETRIEVAL ALGORITHM

6.1 Reducing the set of coefficients

As discussed in the previous chapters, our approach is based on the computation of statistics of detail coefficients. The magnitude of these coefficients depend on both the sharpness of transitions (e.g. edges) in the image and the order of the prediction filter P. Since our database consists of images of single objects, detail coefficients are expected to give a good representation of the contours around and within these objects. Prediction filters of a low order yield a more crisp representation of these contours than high order prediction filters. Obviously, low order filters are bound to match with highly irregular functions while high order filters are related to regular functions/smooth surfaces. As mentioned already before, a prediction filter of low order is very undesirable if the single object is placed on a background of texture and we want to identify similar objects at different backgrounds. Texture can consist of a smooth surface with sharp transitions superposed on it. When using a low order filter all these transitions are translated into detail coefficients with high magnitudes. A high order filter takes a large region around these transitions

into account, see Table 1. Therefore the magnitudes of the coefficients related to the edges in the texture drop.

In order to get a good representation of the crisp object we only use the locally dominant coefficients, related to the object, for computing the moment invariants. A way to do this is to compute for each detail coefficient an approximate value for its gradient. This gradient based information is used to select the coefficents that are likely to be related to the object an not to the background. This is done by means of a threshold value.

So, for a given image I the detail coefficients we first compute the matrices consisting of detail coefficients after each lifting step, i.e., $d_1, \ldots, d_K$. At each level j the small set of coefficients W_j that are used to compute the moment invariants are given by the following rule

$$d_j(n,m) \in W_j \iff |\nabla d_j(n,m)| \geq \sqrt{\sum_{n=1}^{N}\sum_{m=1}^{M} |\nabla d_j(n,m)|^2 / NM}, \tag{6.1}$$

with N and M respectively the number of rows and columns in the matrix d_j. The gradients are approximated numerically. The threshold value in the right-hand side of (6.1) is known in the literature as root mean-square. By selecting the coefficients in this way we do not only want to avoid the influence of coefficients related to the background we also intend to obtain coefficients in W_j that give a sparse representation of the objects. The reason for doing this will be discussed further in Section 6.3.

The algorithm is not yet complete: we need to take additional precautions, see Sections 6.2 and 6.3.

6.2 Preprocessing

When using our algorithm we have to deal with the fact that in practice most affine transformations do not map an image on one (Cartesian) lattice onto an image defined on the same lattice. Generally, an interpolation filter is used to get a new image, which is defined on the original lattice. It can be shown that if the difference between the image f, that was not defined on the original lattice, and its interpolated version g defined on the lattice, is small, then also the difference in the moments μ_{pq} will be small. More precisely, we have

$$|f(x,y) - g(x,y)| < \delta\,|x|^{-\alpha}\,|y|^{-\beta} \implies |\mu_{pq}(f) - \mu_{pq}(g)| < \varepsilon,$$

for $0 \leq p < \alpha - 1$ and $0 \leq q < \beta - 1$. So, in a certain sense, the moments are continuous in the image functions.

Applying a high pass filter (prediction filter) on both the interpolated and the original image yield some other results. Although the detail coefficients of both images will only slightly differ from each other, they will induce large differences in the central moments μ_{pq}. This is due to the fact that the moments are computed out of the magnitudes of these coefficients. So, relatively small differences are not canceled out against each other, but they accumulate towards large differences in the moments. Simulation results have shown that they are highly influenced by this phenomenon.

As a remedy we propose to use a smoothing filter on all images in the database at each resolution level. In this way small blurring filters are used at the lowest resolution level and large blurring filters are used at the highest levels. In our setup a (5×5) 2D cubic spline filter has been used, but other smoothing filters (Gaussian) gave a good performance as well.

6.3 Similitude invariance revisited

In Section 5.2 we have already seen that uniform dilations (by $\alpha > 0$) result in a multiplication of Hu's moments by $\mu'_{pq} = \alpha^{p+q+2}\mu_{pq}$. However, if the object in an image can just be replaced by an orbit on a neutral background, e.g., a circle, then a uniform dilation yields $\mu'_{pq} = \alpha^{p+q+1}\mu_{pq}$. Assuming that the selected detail coefficients also induce orbits instead of regions, this change in moments also holds for the moments computed from these coefficients. To increase stability in the lifting approach all moments were also divided by L_j yielding new moments ν^j_{pq}. For these moments we have

$$\nu^j_{pq} = \alpha^{p+q}\nu^j_{pq}.$$

Following the computations in Section 5.2 invariance of the wavelet moments ν_{pq}^j is achieved by dividing them by $(\nu_{20}^j + \nu_{02}^j)^{\frac{p+q}{2}}$, yielding at level j the new moments

$$\xi_{pq}^j = \frac{\nu_{pq}^j}{(\nu_{20}^j + \nu_{02}^j)^{\frac{p+q}{2}}} = \frac{\mu_{pq}^j \, L_j^{\frac{p+q}{2}-1}}{(\mu_{20}^j + \mu_{02}^j)^{\frac{p+q}{2}}}.$$

We observe that invariant I_1 is now always equal to 1 due to this normalization. Therefore we replace this invariant by $I_1 = \mu_{00}$.

Of course the assumption that the detail coefficients are only grouped into orbits and not into regions is not always justified.

6.4 Weighted Euclidean distance

We have already introduced several (nonlinear) ways of scaling the feature vectors, see Section 5.2. Still, it does not make sense to measure the Euclidean distance between feature vectors as is at this point. We need to adapt our metric further to bring the variances of the elements of the feature vector in comparable range. A commonly used technique is to weigh the elements by variances. See Figure 10, which depicts a $m \times n$ matrix F composed of feature vectors. Each row is an observed feature vector (of length n), corresponding

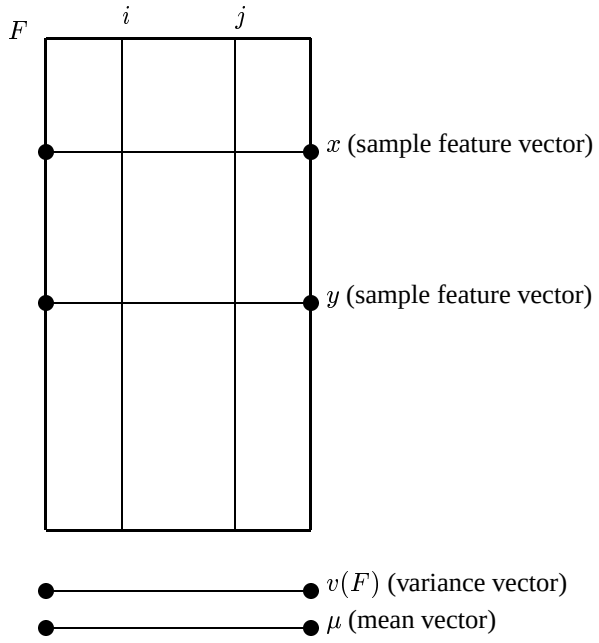


Figure 10: Matrix composed of feature vectors.

to an image. Each column vector corresponds to a specific moment invariant (scaled as in Section 5.2). The column vector has length m, the number of images. The mean vector is the arithmetic average of all rows and the variance of F is defined as a row vector $v(F)$ containing the variance of each column of F. The weighted Euclidean distance is defined as:

$$r_{\text{E}}^2(x, y) \equiv (x - y)^{\text{T}} W^{-1} (x - y) \tag{6.2}$$

where W is a diagonal matrix, the k-th element on the diagonal of which is equal to the k-th element of $v(F)$.

6.5 Mahalanobis distance

The Mahalanobis distance is defined as:

$$r_{\text{M}}^2(x, y) \equiv (x - y)^{\text{T}} C^{-1} (x - y) \tag{6.3}$$

where C is the covariance matrix with respect to F. It is defined as:

$$c_{ij} = \Sigma_{k=1}^{m}(f_{ki} - \mu_i)(f_{kj} - \mu_j)/(n-1), \quad i,j \in \{1,\ldots,n\} \tag{6.4}$$

where f_{ki} is the k,i-th entry of F and μ_i is the i-th element of the mean vector (see Section 6.4). Definitions (6.2) and (6.3) are related to each other in the sense that matrix W is the diagonal matrix of C.

Known advantages of the Mahalanobis metric are that it compensates for an arbitrary scaling of the coordinate axes in the space of feature vectors and that it corrects for correlation between different features. Expression (6.3) has the important property that it is invariant to any nonsingular transformation with matrix $A \in \mathbb{R}^n \times \mathbb{R}^n$. That is, if we substitute each row vector x of F by Ax, then r_{M} remains unchanged, see [10, 9]. The proof hinges on the result that the covariance matrix C is transformed into ACA^{T}.

The covariance matrix C is always symmetric (and positive semidefinite). Therefore C is orthogonal similar to a diagonal matrix (see [16, Thm. 11.3]). It follows that there exists an eigenvalue decomposition

$$C = VDV^{-1} \quad \text{with} \quad V^{-1} = V^{\text{T}}, \tag{6.5}$$

that is, the columns of V are an orthonormal set of eigenvectors. From the above remarks it follows that:

$$r_{\text{M}}^2(x,y) = r_{\text{M}}^2(V^{\text{T}}x, V^{\text{T}}y) = (V^{\text{T}}(x-y))^{\text{T}}\, D^{-1} V^{\text{T}}(x-y). \tag{6.6}$$

This means that if we apply the transformation V^{T} to the feature space, then the weighing of the Mahalanobis metric reduces to weighing through a diagonal matrix D given by (6.5).

6.6 Outline and recapitulation of the retrieval algorithm

Given the many aspects that have been discussed before, we give an outline of the retrieval algorithm. Firstly, we need to build a database with images of interest. In the database each image is accompanied by its feature vector which has to be derived. So-called query images are images which (usually) do not reside in the database and are compared to the elements of the database. Also from a query image a feature vector has to be derived to make comparison possible. Below follows a recap for the construction of a feature vector.

From image to feature vector:

1. Application of a smoothing filter to the image (preprocessing, see Section 6.2).
2. Computation of the relative local variance (RLV) at each point (see (3.1)–(3.2)).
3. For each point a prediction stencil is chosen. The choice depends on the relative local variance (step 2). Where the RLV is low a high order prediction is adopted and where the RLV is high a low order prediction is adopted.
4. Computation of the detail coefficients by the (redundant) lifting scheme. It involves adaptive filtering (see Section 3, adaptivity is steered by step 3).
5. Selection (optional) of the detail coefficients, based on e.g. edge detection.
6. Computation of the moments up to third order with respect to the coefficients resulting from step 4 (see Section 5).
7. Composition of a feature vector based on Hu's moment invariants.
8. Normalization of the feature vector by application of the homogeneity condition (see Section 5.2).
9. Application of the smoothing filter of step 1 to the approximation coefficients.
10. Repetition of steps 2 – 9 up to K levels.
11. Composition of one feature vector from the K feature vectors resulting from step 8.

Once the database has been built, we can compute coefficient matrices C (see (6.3)) or W (see (6.2)) by which distance measures are defined. The distance between images is identified with the distance between the respective feature vectors. For a given query image we find the image in the database which is at the smallest distance. This concludes the retrieval algorithm.

Figure 11: Object library of 8 images of size 128×128.

7. RESULTS

7.1 A synthetic database

To give a 'proof of principle' of our proposed adaptive lifting approach, we constructed a synthetic database of 64 images. These images can be divided into 8 classes, each one consisting of images of one of the objects depicted in Figure 11, but subjected to translation over various distances, reflection over various angles, rotation and the similitude transform (zooming in). Moreover, the images are pasted on an arbitrarily chosen (out of 4) wooden background texture (256×256). For the simulation each image was used as a query to retrieve the other 7 relevant ones.

7.2 The effectiveness of adaptivity

The effectiveness of our approach (solid line) is shown in Figure 12 with both the ideal case (crosses) and the case in which the lifting scheme with a fixed prediction filter was used (dotted line). In this figure the

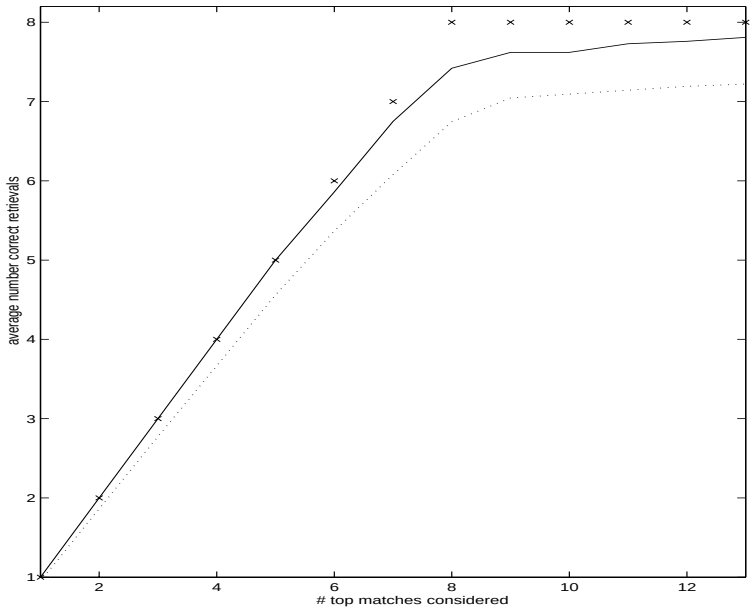


Figure 12: Retrieval performance of adapted (solid) and non-adapted (dotted) approaches.

performance using an 8th order filter has been depicted since it performed slightly better than lifting with

low order filters. Distances among images are measured by weighted Euclidean distances (see Section 6.4) between the feature vectors. The average number of retrieved images of the same class of the query image (vertical axis) has been plotted against different number of allowed top retrievals. As we can see, retrieval rates increase by 5-10% by using an adaptive approach in our test case.

7.3 Euclidean vs. Mahalanobis distance

Obviously, retrieval performances may depend on our definition of distance between images. See Figure 13 for images of distance matrices with respect to the synthetic database. Black denotes the zero value. White

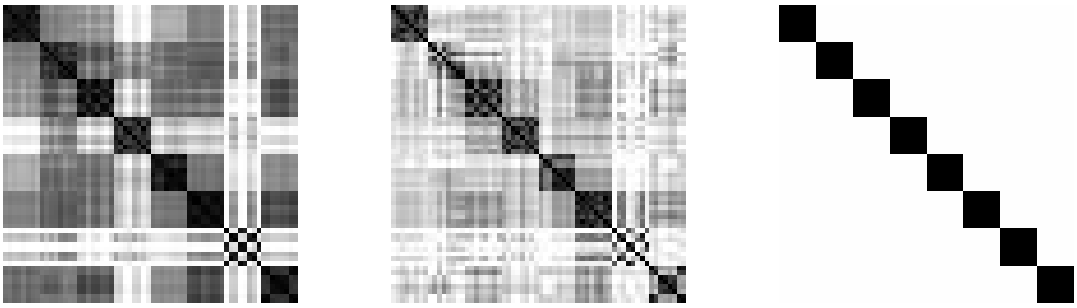


Figure 13: Distance matrices: weighted Euclidean (left), Mahalanobis (middle), ideal (right).

corresponds to the sum of the mean value and the standard deviation of the set of distances, values larger than this sum are depicted as white as well. This choice for the grayscale shows clearly which images are seen as similar by the retrieval algorithm and which are not. For the above synthetic database the image on the right shows the ideal outcome. One of the images in the database has edges that do not contrast well with the background texture, this leads to the poor result in the last but one block on the main diagonal.

The picture suggests that the Mahalanobis distance appears to lead to better discrimination of the images. Nevertheless, retrieval performances measured as in Figure 12 appear to deteriorate somewhat (result not shown).

8. Conclusions/Future Research

In this paper we described how adaptive lifting can be used for certain types of content based image retrieval. Test results show that the adaptive approach performs better than non-adaptive approaches both in a qualitative and a quantitative manner. The combination of adaptive lifting coefficients and moment invariants yields an improved retrieval system based on shape related information. Presumably the adaptive scheme can also be used in combination with other methods to extract other types of features in an image. Besides, following quantitative results we may expect that our approach can also work out for database of objects against many other types of background textures.

Furthermore, we have seen that the classical method of computing moment based invariants encounters some serious difficulties when applying them on images, that have been filtered by a high pass filter (detail coefficients). Both the computations of the moments themselves as well as certain invariants have to deal with these problems. In this paper we suggested solutions to these problems, but further research has to be carried out for a better modeling of these problems.

References

1. W. Ashley, “What shoe was that? The use of computerised image database to assist in identification”, *Forensic Science International*, **82**, 7–20, 1996.
2. R. Baraniuk, R. Claypoole, G. Davis and W. Sweldens, “Nonlinear wavelet transforms for image compression via lifting”, *to appear in IEEE Trans. Imag. Proc.*
3. E. Candes and D. Donoho, “Curvelets, Multiresolution Representation, and Scaling Laws”, *Proc. SPIE 4119, Wavelet Applications in Signal and Image Processing VIII*, 1–12, 2000.
4. T. Chan and H. Zhou, “Adaptive ENO-wavelet transforms for discontinuous functions”, Tech. Rep. 21, Comp. and Appl. Mathematics, UCLA, Los Angeles, 1999.

5. A. Cohen, I. Daubechies and J. Feauveau, “Bi-orthogonal bases of compactly supported wavelets”, *Comm. Pure Appl. Math.*, 45, 485–560, 1992.
6. I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps”, *J. Fourier Anal. Appl.*, 4(3), 345–267, 1998.
7. M. Do, S. Ayer and M. Vetterli, “Invariant image retrieval using wavelet maxima moment”, *Proc. Visual ’99*, Amsterdam, 1999.
8. D. Donoho, “Wedgelets: nearly minimax estimation of edges”, Tech. rep. Statistics Dept. Stanford University, 1997.
9. R.O. Duda, “Pattern Recognition for HCI”, 1996–1997. `www.engr.sjsu.edu/~knapp/HCIRODPR/PR_home.htm`
10. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons , New York, 1973.
11. M. Hu, “Visual pattern recognition by moment invariants”, *IRE Trans. Inf. Th.*, IT-8, 179–187, 1962.
12. J. Kovacevic and W. Sweldens, “Wavelet families of increasing order in arbitrary dimensions”, *IEEE Trans. Imag. Proc.*, 9(3), 480–496, 2000.
13. S. Mallat and Z. Zhang, “Matching Pursuits with Time Frequency Dictionaries”, *IEEE Trans. Signal Processing*, 41(12), 3397–3415, 1993.
14. B. Manjunath and W. Ma, “Texture features for browsing and retrieval of image data”, *IEEE Trans. PAMI*, 18(8), 837–842, 1996.
15. G. Nason and B. Silverman, “The stationary wavelet transform and some statistical applications”, in *Wavelets and Statistics* (Antoniadis and Oppenheim eds.), 281–299, 1995.
16. Evar D. Nering, *Linear Algebra and Matrix Theory*, second edition, John Wiley & Sons , New York, 1970.
17. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer, New York, 1980.
18. W. Sweldens, “The lifting scheme: A construction of second generation wavelets”, *SIAM J. Math. Anal.*, 29(2), 511–546, 1997.
19. W. Trappe and K. Liu, “Adaptivity in the lifting scheme”, *Proc. 33th Conf. on Inf. Sciences and Systems*, Baltimore, 950–955, 1999.
20. G. Uytterhoeven and A. Bultheel, “The red-black wavelet transform”, TW Report 271, Dept. Comp. Sc., Katholieke Universiteit Leuven, Leuven, 1997.
21. A. Wegner, “Global Retrieval, Access and information System for Property items”, 1996. `www.arttic.com/grasp/`